

問い合わせ対応を、 全社の成長エンジンへ

VtigerCRMで始めるカスタマーサポート改革

CSの効率化・品質向上から、営業連携、全社統合型の顧客基盤へ

ONE CUSTOMER ONE HISTORY

01 CS
対応を整える

02 SALES
機会につなぐ

03 COMPANY
経営に活かす

CSの停滞は、情報の分断から生まれる

問い合わせが増えるほど、探す・確認する・引き継ぐ作業も増えてしまう

受付

入口がばらばら

メール、電話、Webフォーム、口頭依頼
が別々に管理される

取りこぼし / 二重対応

対応

背景が見えない

顧客、契約、製品、過去対応を複数の場
所から探す

初動遅延 / 属人化

改善

集計が後追い

分類や原因が揃わず、同じ問い合わせ
が繰り返される

予防できない / 学べない

一元化なしに、標準化・自動化・継続改善は定着しない

顧客・契約・製品・問い合わせを一つにつなぐ

VtigerCRMを、対応記録の箱ではなく「顧客を理解する共通基盤」にする

問い合わせの入口

- ・メール / Webフォーム
- ・電話 / 訪問 / 社内依頼
- ・カスタマーポータル
- ・既存データの取込み



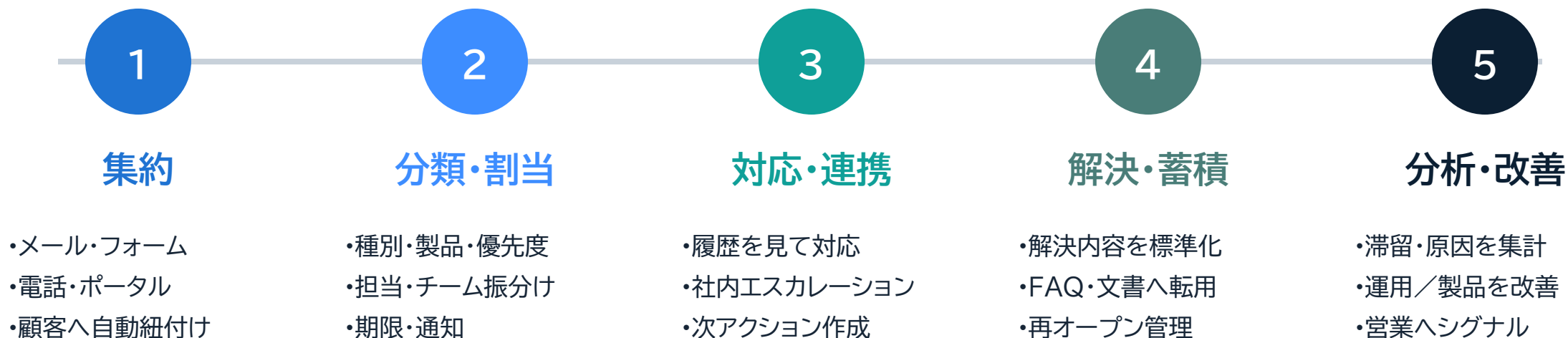
部門ごとの活用

- ・CS:迅速・均質な対応
- ・営業:商談・更新へ連携
- ・開発:再発原因を改善
- ・経営:顧客価値を可視化

顧客・担当者 | チケット | 製品・資産 | サービス契約 | 活動 | FAQ | 文書 | レポート

受付から分析までを、一つの運用サイクルへ

手作業を置き換えるだけでなく、次回の問い合わせを減らすところまで設計する



標準モジュールを軸に、ワークフロー・通知・レポートを業務ルールへ合わせる

高度なSLA、外部チャネル、承認などは拡張・連携・個別開発で補完

「探す・聞く・転記する」時間を顧客対応へ

現場が入力する理由を、管理のためではなく「自分の仕事が楽になる」ことに置く

これまで

- × 複数画面を開いて履歴を探す
- × 経験者に過去の経緯を聞く
- × 同じ内容をメールと表へ転記
- × 担当と次の一手が曖昧

VtigerCRM導入後

- ✓ 顧客画面で、契約・製品・対応履歴を確認
- ✓ 類似チケット・FAQ・解決履歴を参照
- ✓ テンプレートとワークフローで記録・通知
- ✓ 所有者・期限・ステータス・次アクションを明示

初動の安定 | 引継ぎの容易化 | 対応漏れの抑制 | 新人の早期立ち上がり

件数だけでなく、詰まりの原因まで見える

集計のための報告を減らし、日次・週次・月次で意思決定する情報を揃える

頻度	見る指標	決めること
日次	未割当・期限超過・高優先度	誰が、何を、いつまでに対応するか
週次	滞留・担当負荷・エスカレーション	負荷調整、支援、対応ルールの見直し
月次	分類・製品・原因・再発傾向	FAQ、教育、製品・サービス改善の優先順位

見るべき3層

- 01 応答

新規件数 / 初回応答 / 期限超過
- 02 解決品質

解決時間 / 再オープン / 上位連携
- 03 予防

頻出分類 / 発生原因 / FAQ活用

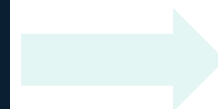
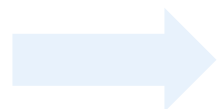
ダッシュボードは「眺める画面」ではなく、会議で次の行動を決める画面として設計します

ポータルとFAQで、自己解決と透明性を高める

顧客自身が必要な情報へ到達でき、問い合わせは最初から構造化される

顧客ができること

- ・問い合わせ登録
- ・状況・回答の確認と更新
- ・ファイルの共有
- ・FAQ・ナレッジ検索
- ・プロフィール情報の確認



CS側で得られる効果

- ・定型問い合わせの削減
- ・受付項目・分類の標準化
- ・やり取りの自動集約
- ・回答状況の透明化
- ・ナレッジの継続改善

認証・権限・公開情報・対象モジュールは、OSS標準／拡張／個別開発の範囲を要件定義で確定

CSの「顧客の声」を営業の一次情報へ

担当者の記憶に頼らず、チケットの内容から次アクションを確実に起こす

CSで見えるシグナル	営業への連携	狙い
追加利用・新要件の相談	案件候補を作成し、営業担当へタスクと通知	アップセル / クロスセル
不満・利用停滞・解約示唆	更新前フォロー、責任者確認、対応期限を設定	解約リスクの早期対応
故障・消耗・更新時期の兆候	保守・交換・更新提案のタイミングを共有	適時提案と顧客負荷の軽減
高評価・成功事例の兆候	事例化、紹介依頼、他部門展開を営業へ連携	関係拡大と信頼の活用

ルール例:分類・優先度・製品・キーワード・契約期限を条件に、案件／活動／通知を自動作成

顧客単位的全履歴が、部門間の引継ぎをなくす

各部門が別々の表を持つのではなく、同じ顧客レコードを権限に応じて利用する



部門をまたぐ共通キーは「顧客」。案件・契約・問い合わせ・請求まで、同じ事実を共有します。

問い合わせデータを、三つの経営成果へ

CSはコストセンターではなく、継続・拡大・改善を生み出す顧客接点になる

01

継続率を守る

不満・滞留・再発
↓
リスクを早期共有
↓
先回りしたフォロー

02

取引を広げる

相談・利用変化
↓
案件・活動へ変換
↓
適切な提案

03

提供価値を高める

頻出問い合わせ
↓
原因と影響を集約
↓
製品・業務を改善

顧客ライフサイクル

獲得 → 営業 → 導入・提供 → サポート → 更新・拡大

CS起点の三段階で、成果を全社へ広げる

最初から全社最適を作り込まず、CSで使える最小構成から段階的に拡張する

PHASE 1

CS基盤

- ・顧客・担当者データ
- ・受付チャンネルとチケット
- ・分類・状態・担当・期限
- ・通知と基本レポート

対応漏れと滞留が見える化

1

PHASE 2

営業連携

- ・案件・活動履歴
- ・CSシグナルの自動連携
- ・見積・提案情報の共有
- ・更新・解約リスク管理

顧客の声を次の行動へ

2

PHASE 3

全社統合

- ・契約・製品・資産
- ・受注・請求・プロジェクト
- ・外部システム連携
- ・経営ダッシュボード

顧客情報を全社資産化

3

範囲・期間は、利用人数、データ移行、外部連携、承認・セキュリティ要件により調整します。

OSSの柔軟性を、運用できる仕組みに変える

利用部門を広げやすい土台と、継続利用を支える運用設計をセットで考える

拡張しやすい基盤

ユーザー課金に縛られにくい

OSS版VtigerCRMのソフトウェア利用は、一般的なSaaSのような利用者単位の月額課金ではありません。

業務に合わせて変えられる

項目、画面、ワークフロー、レポート、カスタムモジュール、API連携を段階的に追加できます。

費用を共通基盤へ投資できる

ライセンス中心ではなく、構築・連携・運用・改善へ配分しやすくなります。

安心して使い続ける運用

- | | | |
|----|----------|-------------------|
| 01 | AWS環境設計 | 可用性・性能・バックアップ方針 |
| 02 | セキュリティ運用 | 更新、監視、権限、アクセス制御 |
| 03 | 変更管理 | 拡張互換性とテスト環境での検証 |
| 04 | CRM管理者支援 | 設定変更、データ品質、利用定着 |
| 05 | 継続改善 | KPIと現場フィードバックで見直し |

※ ソフトウェアライセンスが無償でも、サーバー、構築、保守、拡張機能、セキュリティ運用等の費用は必要です。

まず90日を目安に、 動くCS基盤を定義する

- 1 現状整理** 入口、件数、分類、担当、期限、KPIを整理
- 2 最小設計** 項目・状態・ワークフロー・権限・レポートを定義
- 3 試行と改善** 代表ケースを一チームで検証し、運用へ反映

PHASE 1 DELIVERABLES

初期構築で 決めるもの

- ・CS業務フロー
- ・VtigerCRM設定一覧
- ・移行データと連携方針
- ・運用ルールとKPI
- ・全社展開ロードマップ

CSの効率化は入口。顧客情報を全社の共通資産に変える。